

CHAPTER 4-RESULTS

4.1 Direct Communication Mode

In this section, the results of the **Direct Communication Mode** are discussed. In this mode, all nodes share data with each other directly without any intermediate routing. A node accurately identifies and displays data received from other nodes in real-time in addition to sending its own data. In the following each node is connected to a laptop, displaying results on Arduino IDE software.

4.1.1 Channel State Information

- **Channel between Node 1 and other Nodes:**

The beacon message broadcast from this node is displayed at the top. Afterward, the channel state information of a packet received from a nearby Node 2 is displayed. Signal-to-noise ratio, link quality using the received signal's strength(Signal written in image below) , and data parameters are processed and shown. The received signal strength (RSSI) of Node 1 from Node 2 is -58 dBm, which is a strong and stable wireless link. A greater (less negative) RSSI value indicates little signal loss through the channel, guaranteeing successful communication. This reading implies Node 2 is near or within a good transmission range greater (less negative) RSSI value indicates little signal loss through the channel, guaranteeing successful communication. This reading implies Node 2 is near or within a good transmission range

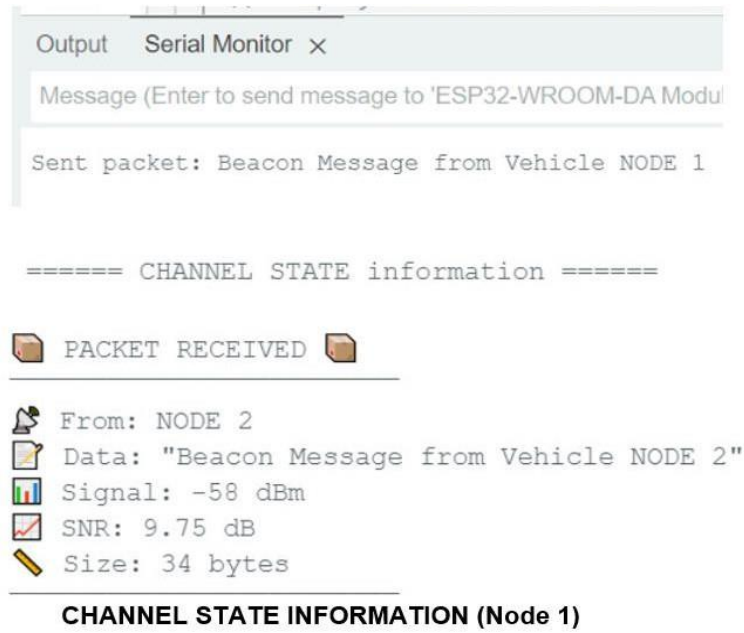


Figure 16 Channel state information (node 1)

In the similar way packets from Node 3 and Node 4 have been received and processed by this node.

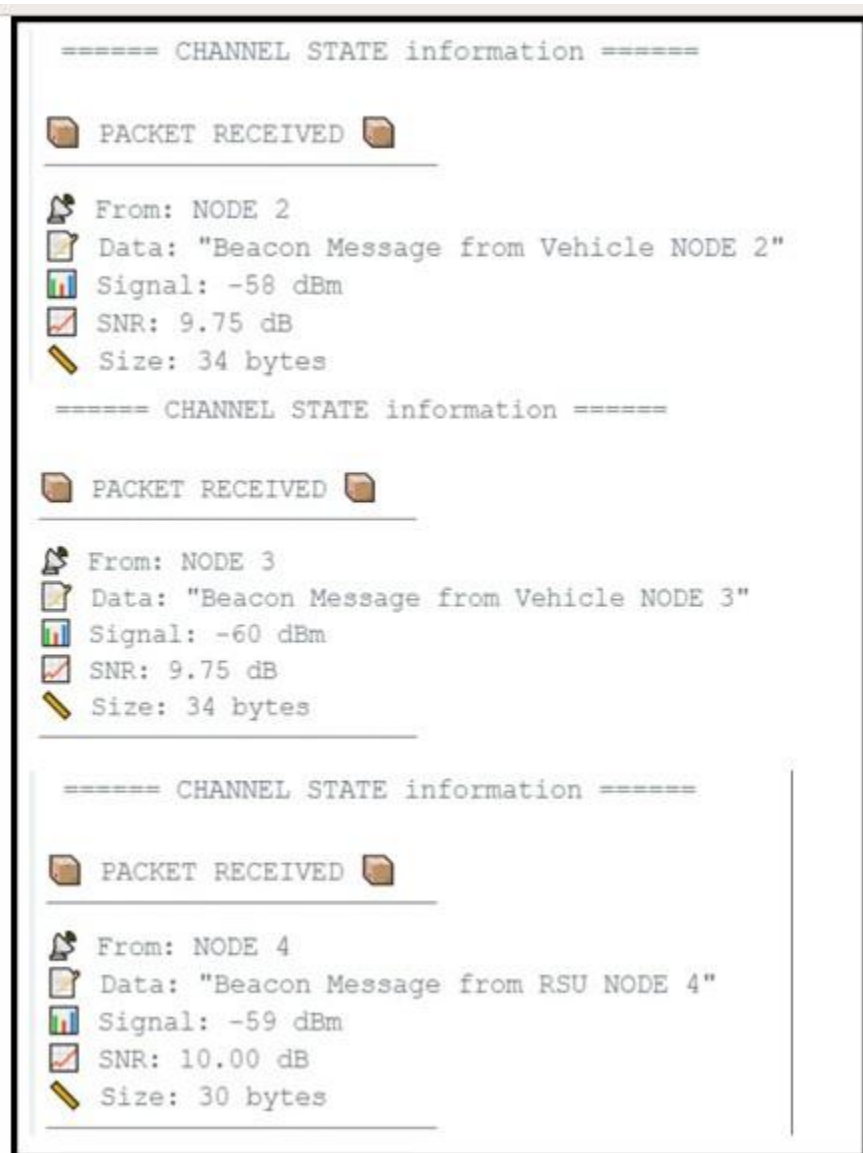


Figure 17 Channel state information (node 2, node 3, node 4)

- **Channel between Node 2 and other Nodes:**

Similarly at Node 2, the packets are being received, processed, and accurately displayed in real-time for network assessment. Also the packet broadcasted by it is displayed as well.

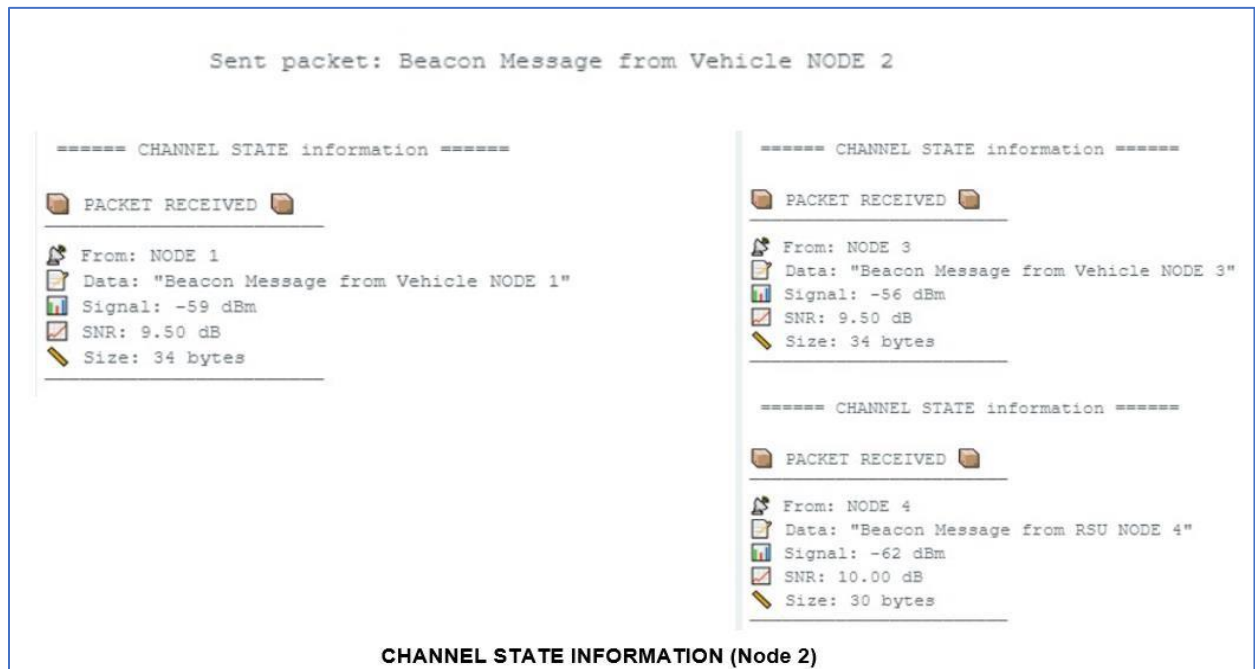


Figure 18 Channel state information(node 2)

- **Channel State Information of Node 3 and Node 4:**

The channel state information of Node 3 and Node 4 are displayed respectively:

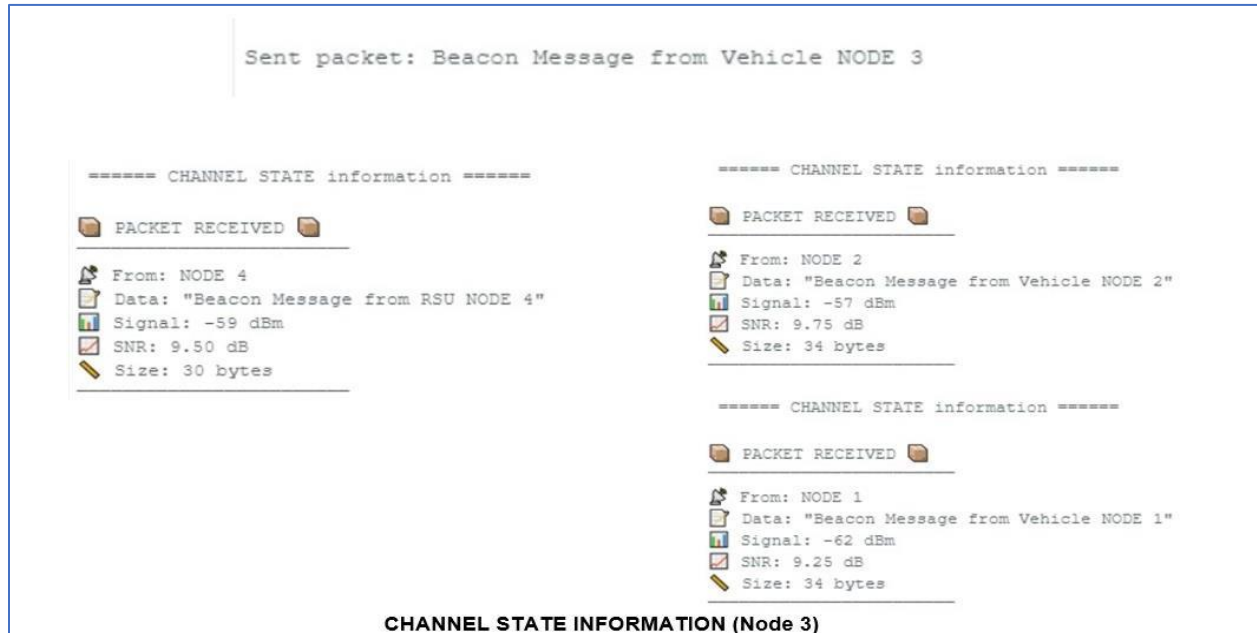


Figure 19 Channel State information (node 3)

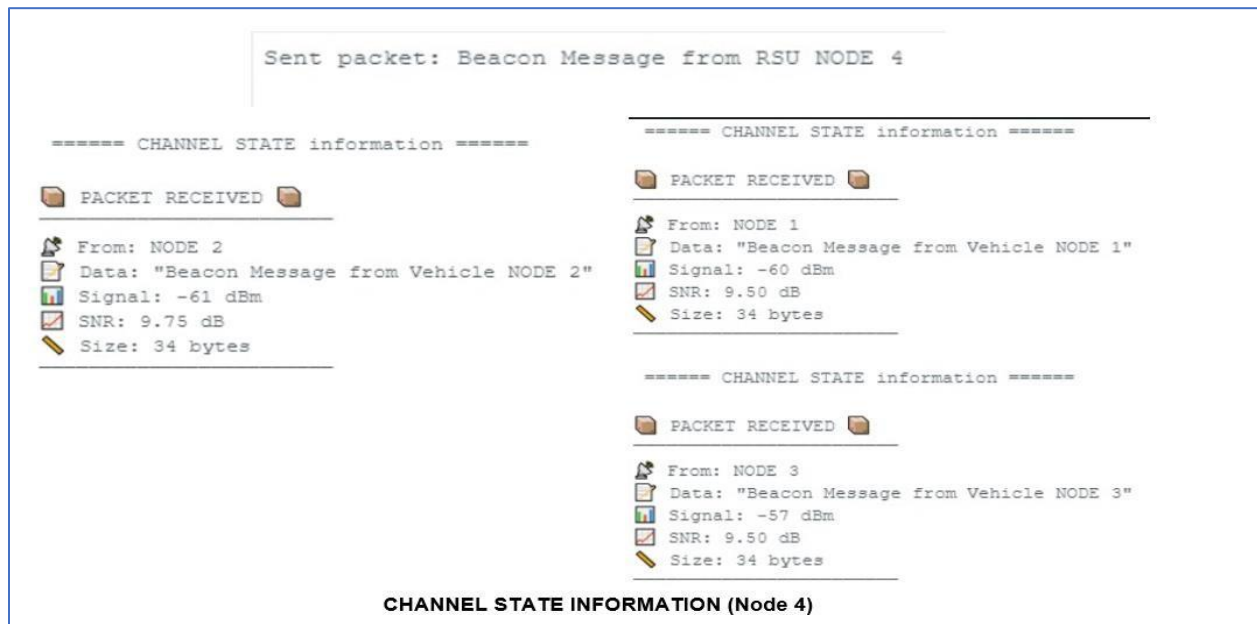


Figure 20 Channel State information(node 4)

4.1.2 Node State Information

- **Node 1**

In node state information, the distance of a node from which a packet is received is calculated and displayed. Last data received and signal strength are also displayed. From this distance, nearest node is determined and shown.

```
===== Current NODE STATE Information =====
Current Node: NODE 1
📶 Nearest Node: NODE 2 (Distance: 7.1m)

Data recieved from vehicles:

From: "NODE 3"
Last Data: "Beacon Message from Vehicle NODE 3"
Signal Strength: -60 dBm
Total Packets: 4

-----

From: "NODE 4"
Last Data: "Beacon Message from RSU NODE 4"
Signal Strength: -60 dBm
Total Packets: 4

-----

From: "NODE 2"
Last Data: "Beacon Message from Vehicle NODE 2"
Signal Strength: -57 dBm
Total Packets: 4

-----

Sent packet: Beacon Message from Vehicle NODE 1
```

Figure 21 Node state information (node 1)

- **Node 2:**

Similarly the Node state information of Node 2 has been displayed. Nearest node, signal strength, and data received from other nodes is extracted from the received packets and shown.

===== Current NODE STATE Information =====

Current Node: NODE 2

📶 Nearest Node: NODE 3 (Distance: 5.6m)

Data recieved from vehicles:

From: "NODE 1"

Last Data: "Beacon Message from Vehicle NODE 1"

Signal Strength: -61 dBm

Total Packets: 5

From: "NODE 4"

Last Data: "Beacon Message from RSU NODE 4"

Signal Strength: -60 dBm

Total Packets: 6

From: "NODE 3"

Last Data: "Beacon Message from Vehicle NODE 3"

Signal Strength: -55 dBm

Total Packets: 5

Sent packet: Beacon Message from Vehicle NODE 2

NODE STATE INFORMATION (Node 2)

Figure 22 Node state information (node 2)

- **Node 3 and Node 4:**

```

===== Current NODE STATE Information =====
Current Node: NODE 3
📶 Nearest Node: NODE 2 (Distance: 6.3m)

Data recieved from vehicles:

From: "NODE 4"
Last Data: "Beacon Message from RSU NODE 4"
Signal Strength: -58 dBm
Total Packets: 5

From: "NODE 2"
Last Data: "Beacon Message from Vehicle NODE 2"
Signal Strength: -56 dBm
Total Packets: 5

From: "NODE 1"
Last Data: "Beacon Message from Vehicle NODE 1"
Signal Strength: -62 dBm
Total Packets: 5

Sent packet: Beacon Message from Vehicle NODE 3

```

NODE 3

Figure 23 Node state information (node 3)

```

===== Current NODE STATE Information =====
Current Node: NODE 4
📶 Nearest Node: NODE 3 (Distance: 7.1m)

Data recieved from vehicles:

From: "NODE 2"
Last Data: "Beacon Message from Vehicle NODE 2"
Signal Strength: -59 dBm
Total Packets: 6
-

From: "NODE 3"
Last Data: "Beacon Message from Vehicle NODE 3"
Signal Strength: -57 dBm
Total Packets: 7
-

From: "NODE 1"
Last Data: "Beacon Message from Vehicle NODE 1"
Signal Strength: -58 dBm
Total Packets: 7
-

Sent packet: Beacon Message from RSU NODE 4

```

NODE 4

Figure 24 Node state information (node 4)

4.1.3 Road Side Unit prototype Node 4

A USRP-B210 has been attached with the same laptop with Node 4 connected:

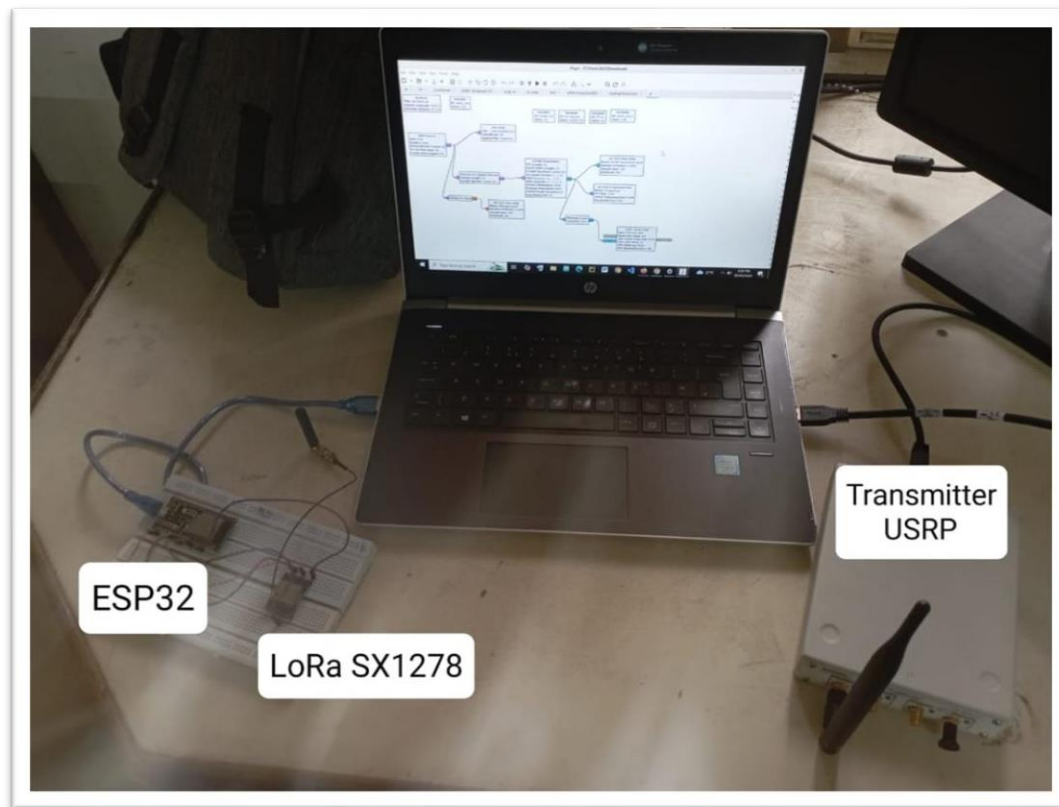


Figure 25 Setup for Node 4 with USRP attached

For this we have placed the three nodes at close distance in order for them to send their respective sensor data (GPS coordinates and Acceleration). Since these nodes are in close proximity of Node 4, the sensors data at the vehicle nodes will be fetched and send instead of a beacon message.

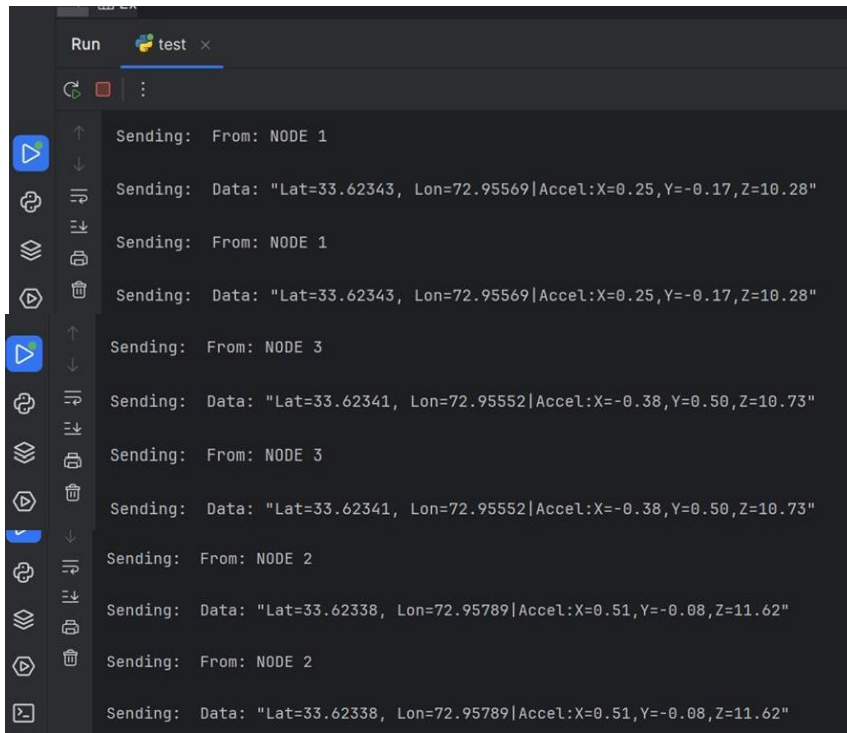
A script of UDP Socket running on the laptop takes data from COM Port 7 to Port 9600. This 9600 port address is defined in the UDP Source block of OFDM transmitter block in GNU Radio.

4.1.3 Node 4 Data Transmission and Reception via OFDM

4.1.3.1 Data Transmission from Nodes 1, 2, and 3 (TX Side)

- **Data Transferred from Node 1, 2, 3 to Node 4 via Python Script:**

Following are the snapshots of the vehicle nodes data transfer to Port 9600 from a python script:



```
Run test x
Sending: From: NODE 1
Sending: Data: "Lat=33.62343, Lon=72.95569|Accel:X=0.25,Y=-0.17,Z=10.28"
Sending: From: NODE 1
Sending: Data: "Lat=33.62343, Lon=72.95569|Accel:X=0.25,Y=-0.17,Z=10.28"
Sending: From: NODE 3
Sending: Data: "Lat=33.62341, Lon=72.95552|Accel:X=-0.38,Y=0.50,Z=10.73"
Sending: From: NODE 3
Sending: Data: "Lat=33.62341, Lon=72.95552|Accel:X=-0.38,Y=0.50,Z=10.73"
Sending: From: NODE 2
Sending: Data: "Lat=33.62338, Lon=72.95789|Accel:X=0.51,Y=-0.08,Z=11.62"
Sending: From: NODE 2
Sending: Data: "Lat=33.62338, Lon=72.95789|Accel:X=0.51,Y=-0.08,Z=11.62"
```

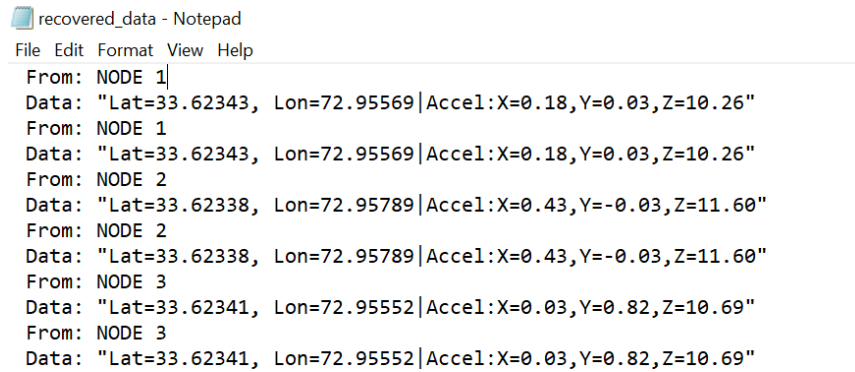
Figure 26 Data Transferred from Node 1, 2, 3 to Node 4 via Python Script

This GPS and Acceleration data of vehicles nodes are received by the UDP Source block in GNU Radio. From there, this data is fed into the OFDM Transmitter block, modulated and transmitted over the air from the USRP Radio.

4.1.3.2 Data Reception and Recovery on Receiver side USRP

Another USRP B210 attached with a laptop receives this ofdm modulated data, demodulates, extracts and stored the original vehicle nodes data into a bin file. The Receiver flow graph is running on the GNU Radio. In this graph after the OFDM block's output is connected to a File Sink block, in which path for an empty file named as recovered_data is provided. This purpose of this is to store the recovered data and verify the results.

- **Recovered Data Stored in Binary File:**



```
recovered_data - Notepad
File Edit Format View Help
From: NODE 1
Data: "Lat=33.62343, Lon=72.95569|Accel:X=0.18,Y=0.03,Z=10.26"
From: NODE 1
Data: "Lat=33.62343, Lon=72.95569|Accel:X=0.18,Y=0.03,Z=10.26"
From: NODE 2
Data: "Lat=33.62338, Lon=72.95789|Accel:X=0.43,Y=-0.03,Z=11.60"
From: NODE 2
Data: "Lat=33.62338, Lon=72.95789|Accel:X=0.43,Y=-0.03,Z=11.60"
From: NODE 3
Data: "Lat=33.62341, Lon=72.95552|Accel:X=0.03,Y=0.82,Z=10.69"
From: NODE 3
Data: "Lat=33.62341, Lon=72.95552|Accel:X=0.03,Y=0.82,Z=10.69"
```

Figure 27 Recovered Data stored in a file

The data is accurately recovered by OFDM receiver and is displayed in the above figure.

4.2 SNR, RSSI AND DISTANCE GRAPHS

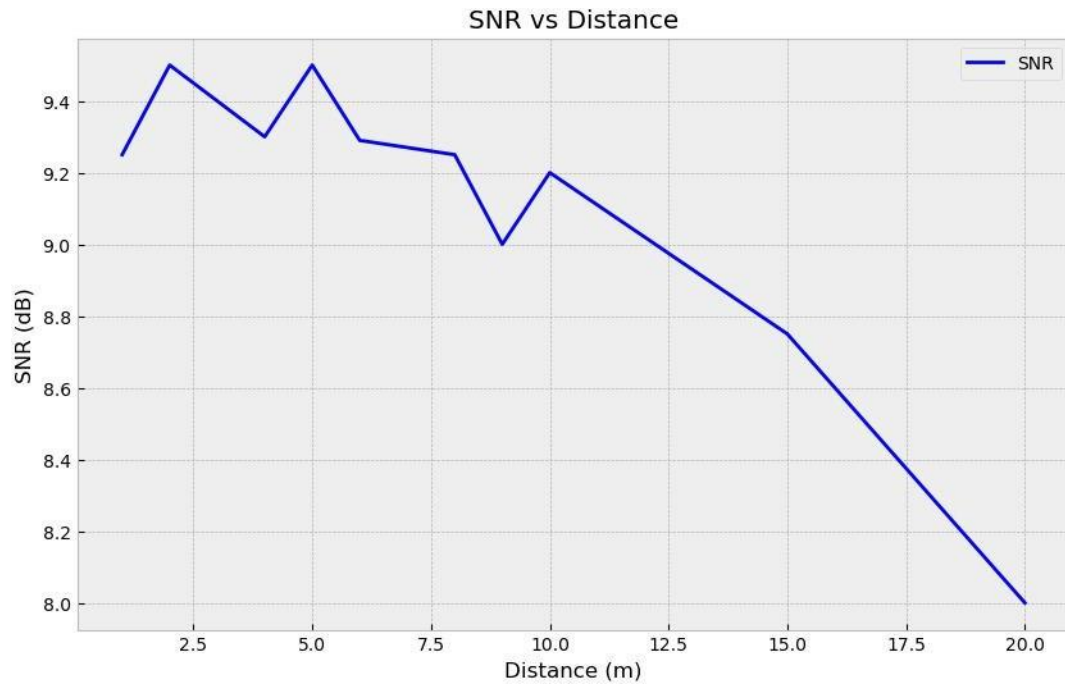


Figure 28 SNR Vs Distance graph

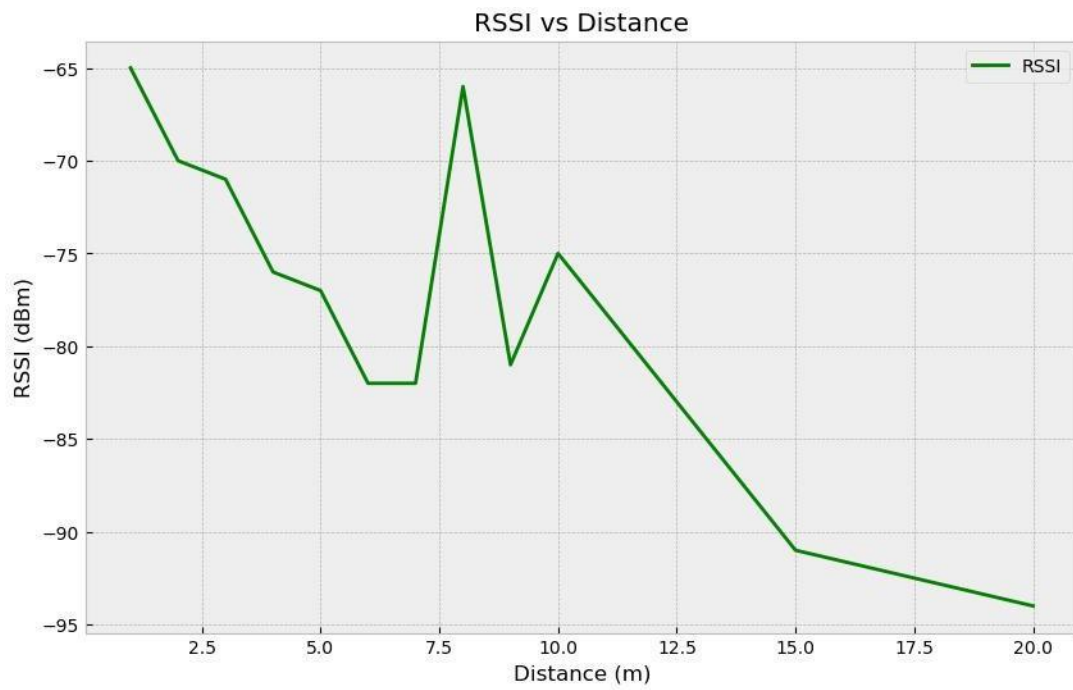


Figure 29 RSSI Vs Distance graph

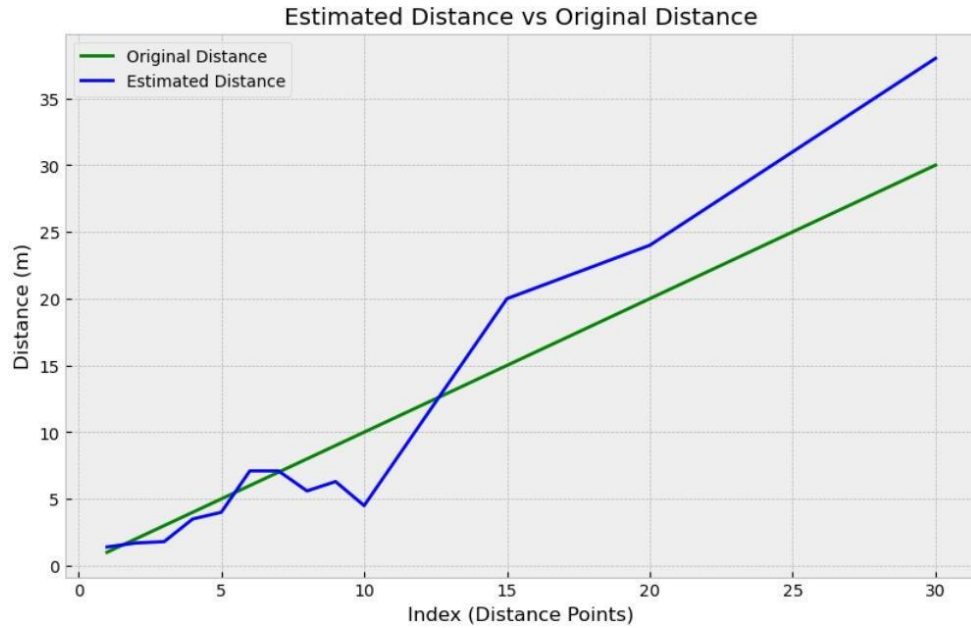


Figure 30 Estimated Distance Vs Actual Distance graph

4.3 Time and Frequency Spectrum Analysis

- **Transmitter Side Spectrum:**

Following are the Time and Frequency domain plots of the transmitter side:

The below graph shows the time domain OFDM signal transmitted over the air. The red and blue graphs are the In-phase and Quadrature components, respectively. These are results of BPSK Modulation on the subcarriers. Original data lies in the In-phase component since the Quadrature component vanishes because of 0 and 180 degrees of phase shift.

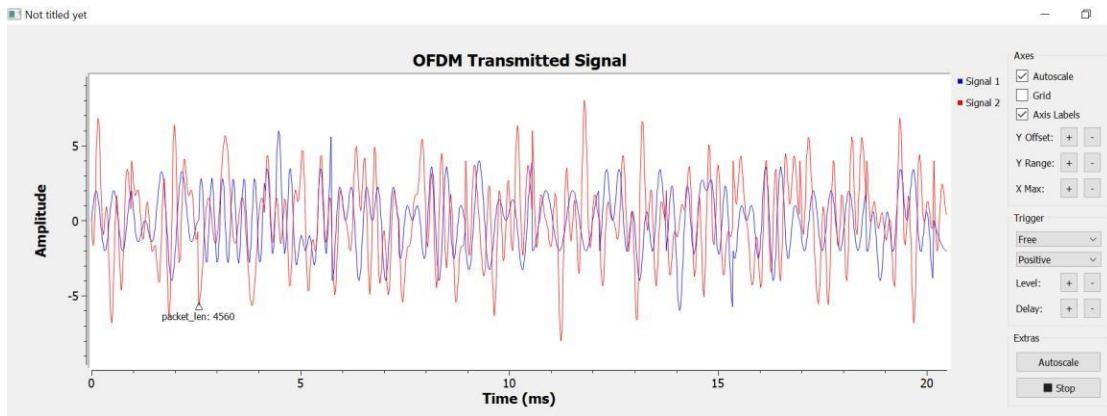


Figure 31 OFDM Transmitted Signal

Below is the graph of Frequency spectrum of OFDM Transmitter, centered around 900MHz over a bandwidth of 20MHz (from **890 MHz to 910 MHz**). The peaks in the signal shows the modulated subcarriers containing the data at specific frequencies.

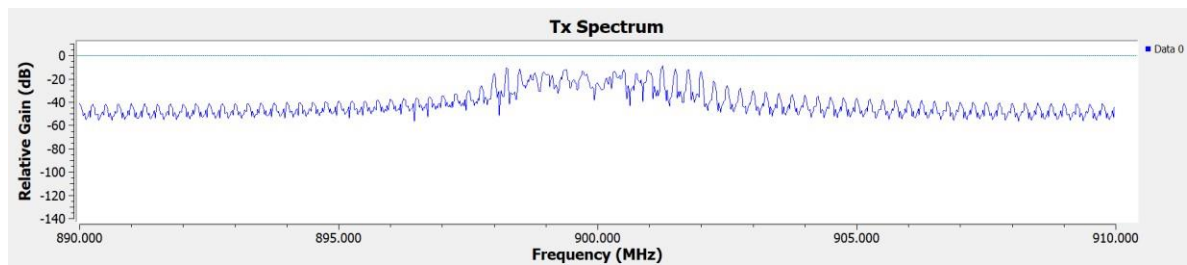


Figure 32 Tx Spectrum

This plot shows the time domain signal of the actual data being transmitted. The variation in the blue graph shows how amplitude of the data changes over time. The sharp peaks and oscillations show that data is transmitted in discrete chunks.

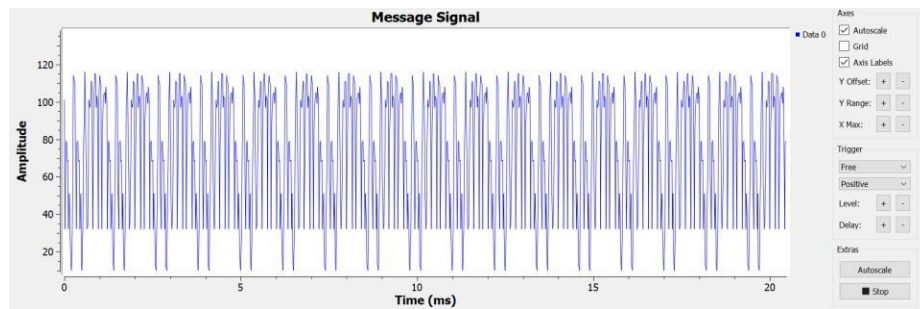


Figure 33 Message Signal

- **Receiver Side Spectrum:**

Following are the receiver side Time and Frequency domain spectrums.

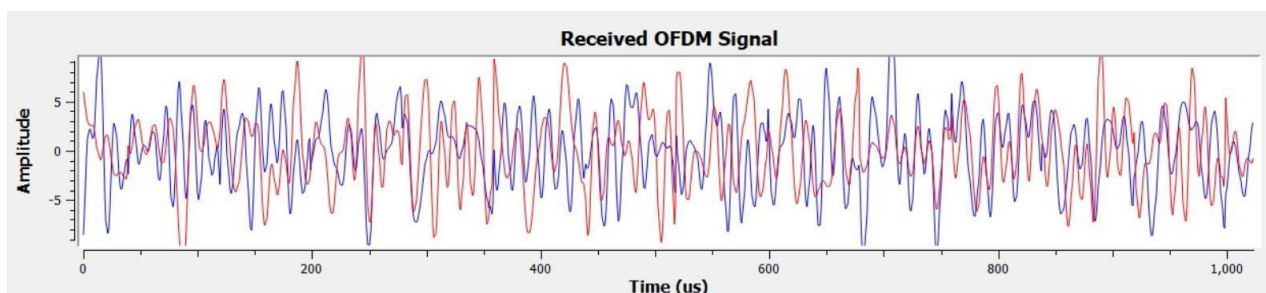


Figure 34 Received OFDM Signal

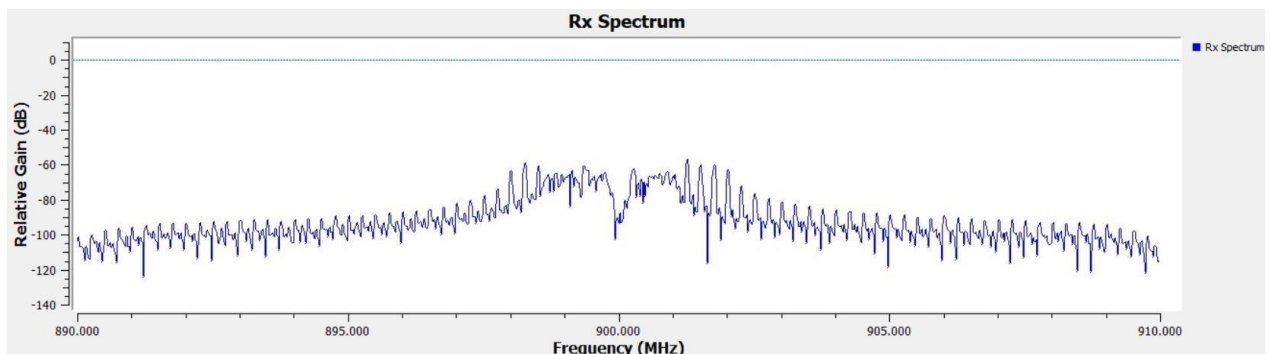


Figure 35 Rx Spectrum

Recovered Message Signal Plot

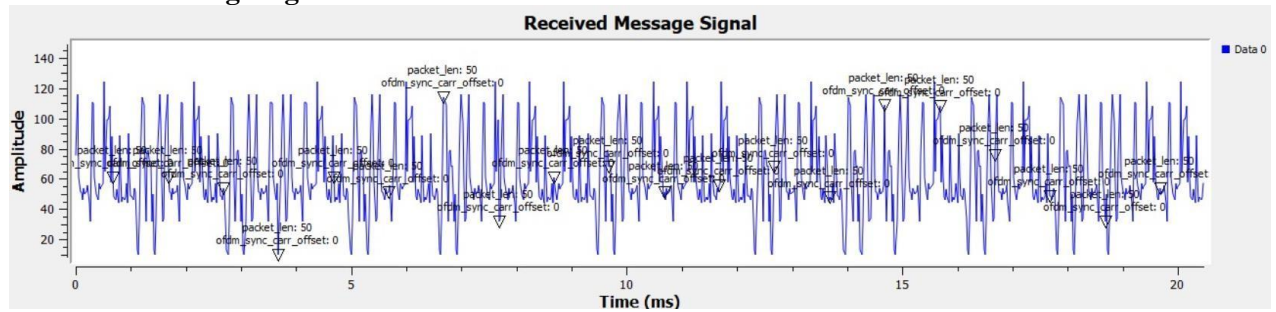


Figure 36 Received Message Signal

The graph shows the amplitude of the recovered data and is similar to the one transmitted from Node 4.

4.4 Dynamic Source Routing Protocol

Following is the setup of nodes, placed apart with each connected to a laptop displaying results on Arduino IDE software.



Figure 37 Nodes setup for implementing DSR

4.4.1 Route Discovery

Source Node 1 initiates a beacon in order to determine a nearest node, to find route to reach the destination node. Node 2 is the nearest node, which receives this packet and generates a beacon reply message.

```
Received Beacon Reply from: Node2 | RSSI: -78
Received Another Beacon Reply from: Node2 | RSSI: -78
Received Another Beacon Reply from: Node2 | RSSI: -77
📌 Nearest Node Selected: Node2
🚀 ROUTE REQUEST SENT 🚀
Source: Node1
Destination: Node4

✅ ACK Received from Node2
```

Figure 38 link established between two nodes

A link is established between the two nodes from thi beacon reply.

```

BEACON sent back successfully
-----
Preparing to send back Beacon Reply
BEACON sent back successfully
-----
Waiting for the packets

```

Figure 39 Node 2 matches the destination id

Node 2 matches the destination id name with its own id, since its not the destination so it finds its next nearby node to forward the route request . A Beacon Reply at this node , from nearby Node 3 is received.

```

Received Another Beacon Reply from: Node3 | RSSI: -86
Received Another Beacon Reply from: Node3 | RSSI: -90
Received Another Beacon Reply from: Node3 | RSSI: -90
📌 Nearest Node Selected: Node3
Forwarded Route Request

```

Figure 40 Route Request is forwarded to Node 3

Route Request is forwarded to Node 3 , since its not the destination node as well, so in the similar way finds its next nearby node, and forwards this received request.

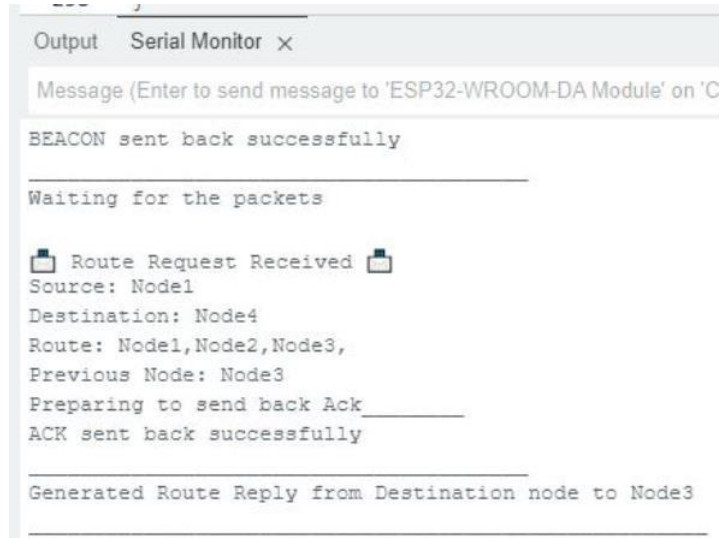
```

📧 Route Request Received 📧
Source: Node1
Destination: Node4
Route: Node1,Node2,
-----
BEACON msg sent successfully
Received Beacon Reply from: Node4 | RSSI: -85
Received Another Beacon Reply from: Node4 | RSSI: -85
Received Another Beacon Reply from: Node4 | RSSI: -86
Received Another Beacon Reply from: Node4 | RSSI: -87
📌 Nearest Node Selected: Node4
Forwarded Route Request

```

Figure 41 request arrives from multihop at Node 4

Finally the route request arrives from multihop at Node 4.



```
Output  Serial Monitor x
Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'C
BEACON sent back successfully

Waiting for the packets

📡 Route Request Received 📡
Source: Node1
Destination: Node4
Route: Node1,Node2,Node3,
Previous Node: Node3
Preparing to send back Ack
ACK sent back successfully

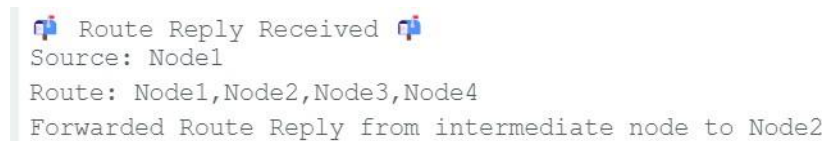
Generated Route Reply from Destination node to Node3
```

Figure 42 At Node 4, the destination id will be matched

At Node 4, the destination id will be matched with the Node name, so it generates a RouteReply on the same path in the reverse order in which request is received.

4.4.2 Route Reply

The destination Node 4 unicasts the route reply packet by appending its own name in the route towards the Node 3. This reply is forwarded by Node 2 and Node 3 and finally route reply packet arrives at Source Node



```
📡 Route Reply Received 📡
Source: Node1
Route: Node1,Node2,Node3,Node4
Forwarded Route Reply from intermediate node to Node2
```

Figure 43 Node 3 forwarding route reply

1.

```
📡 Route Reply Received 📡  
Source: Node1  
Route: Node1,Node2,Node3,Node4  
Generating Route Reply  
  
Forwarded Route Reply from intermediate node to Node1
```

Figure 44 Node 2 forwarding route reply

```
📡 Route Reply Received 📡  
Source: Node1  
Route: Node1,Node2,Node3,Node4  
  
🔦 ROUTE DISCOVERED 🔦  
Full Route: Node1,Node2,Node3,Node4  
Waiting for route replies
```

Figure 45 Route Reply arrived at Node 1

Route Reply arrived at Node 1. After receiving this packet, Source node fetches the sensor data which are GPS longitudes latitudes, and Acceleration in x,y and z axis, prepares a data packet and forwards this on the discovered route hop by hop. Objective is to route this data successfully to the destination Node 4.

```
📡 Data Packet Sent: DATA:PacketDest:Node2:ROUTE:Node1,Node2,Node3,Node4:SOURCE:Node1:DESTINATION:Node4:PAYLOAD:33.622923,72.958043  
Waiting for route replies  
✅ Valid GPS Data Captured: 33.624214,72.957679  
Next node is: Node2  
📡 Data Packet Sent: DATA:PacketDest:Node2:ROUTE:Node1,Node2,Node3,Node4:SOURCE:Node1:DESTINATION:Node4:PAYLOAD:33.624214,72.957679  
Waiting for route replies  
✅ Valid GPS Data Captured: 33.625242,72.957944  
Next node is: Node2
```

Figure 46 Data packet containing the sensor data

This data packet containing the sensor data reaches successfully at the Node 4.

```
Data Packet Received successfully as follows: ✅  
  
DATA:PacketDest:Node4:ROUTE:Node1,Node2,Node3,Node4:SOURCE:Node1:DESTINATION:Node4:PAYLOAD:GPS:Lat=33.622677, Lon=72.957696|Accel:X=0.49,Y=0.20,Z=11.53
```

Figure 47 Dynamic behaviour is verified